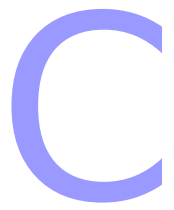
Süsteemprogrammeerimine keeles C



15 Loeng

I18n ja L10n

- Internationalization – keelte võimaldamine
- Localization – tõlkimine ja kohandamine

ASCII

- Varasem aeg meid õnneks ei mõjuta
- ASCII standard: tähemärgid mis on väiksemad kui 32 ei ole trükitavad (kõll ja printerisse uue lehe saatmine jne)
- Tähemärgid üle 127 vabaks kasutamiseks

IBM PC kooditabel (437)

- ASCII ühilduvus
- Mõnede Euroopa keelte é ja è tähed
- Püst ja kaldkriipsud tabelite jaoks
- Meenutagem mis nägu olid kassa-arvutite pildid
  - (Selliste näitamiseks on teek curses)
- Aga heebrea keel,
  Aasia keeled,
  Vene keel?

Illustratsioon: https://en.wikipedia.org/wiki/File:Codepage-437.png

Kooditabelid

- ASCII põhi ja ülemised tähed teiselaadsed
- Mitme keele paraleeltugi

- Aasias samal ajal kahebaidine kodeering
- Tulemus: samas arvutis on võimatu kasutada kaht keelt korraga (kui ise bitmap-fonte ei tee)
- IBMi ja Microsofti kooditabelid hargnevad pärast koostöö lõppu 1990ndatel
- Siia vahele ka ANSI standardi märksõna

Appi tõttab Unicode

- Unicode on oma olemuselt kogu *Code Point* nimelisi objekte
- Iga *Code Point* viitab mingile konkreetsele märgile, mis mõnikord on tähemärk mingis konkreetses keeles, näiteks A või Õ või ffi (U+FB03)
- Neid võib olla sisuliselt lõputus koguses
- Neile *Code Point*'idele võib viidata konkreetsete kodeeringute abil

Funktsionaalsus

- Kombineeruvad tähed
- ~ ja o > õ tüüpi kombineeringud
- test
- Märk mis vahetab teksti paremalt vasakule minema

Näide

- test
- 74 65 73 74
- 2 baidine: 00 74 00 65 00 73 00 74 (UCS-2 / UTF-16)
- Või? : 74 00 65 00 73 00 74 00
- FE FF : byte order mark
  - Keegi Microsoftis arvas et see on hea asi ja käib failide ja stringide ette; Unixi maailmas vältida
- UCS-4 tähendab neljabaidiseid tähemärke

Kodeering: UTF-8

- Üks konkreetne kodeering
- Alumised 127 baiti on ASCII ühilduvad
- Edasised baidid tähistavad mitmebaidilisi tähemärke
- Linux standardiseerus sellele vahepeal ära; ei ole otstarbekas midagi muud kasutada

Järeldused

- Lihttekst on väga suhteline
- Meid huvitab alati selle kodeering
  - Meie pakkija programm oli ju ka sisuliselt ühest kooditabelist teise teisendaja

Praktilist

- GNU teek: libiconv
  http://www.gnu.org/software/libiconv/
- fopen(„fail.txt“, „r, ccs=UTF-8“);
- wchar_t andmetüüp
- fgetc() >> wint_t fgetwc(FILE * stream)
- EOF >> WEOF

Linuxi tugi

- Klaviatuurisisend (terminalist saabuv stdin) muudetakse vastavaks UTF-8 baidijadaks (konsoolidraiveri tasemel)
- Konsooliväljund dekodeeritakse UTF-8 dekoodriga ja esitatakse 16-bitise fondiga
- BOM puudub (too FE FF)

Kaks lähenemist

- Hoida andmeid sisemiselt UTF-8 kodeeringus
- Hoida andmeid lahtivõetud kujul ja konverteerida enne väljastamist
    - Tähemärk oleks siis mingi konkreetne mäluobjekt

Esimese viisi puudused

- strlen() ei ütle mitu positsiooni kursor edasi liigub
- mbstowcs(NULL,s,0) annab vastavalt kodeeringule lõpliku tähemärkide hulga

Kasutamine

- Defineeritud peaks olema locale :
  LANG=et_EE (väljund ISO 8859-1)
  LANG=et_EE.UTF-8 (väljund UTF-8)
- #include <locale.h>
- setlocale() - LC_CTYPE või LC_ALL argumendiga
- käsk:
  locale -a näitab süsteemi paigaldatud piirkonnad

Gettext

- Lahenduse ehitas Sun Microsystems
- Kopeeris GNU projekt
- Üsna standardne ja laialt kasutuses

Töö kulg

- Kirjuta programm arvestades gettext() funktsiooni ja lokaatide registreerimist
- Korja xgettext programmiga stringid .pot faili
- Loo msginit programmiga kohaliku keele tõlkefail
- Tõlgi
- Konverteeri msgfmt programmiga tõlge binaarseks
- Paiguta tõlge /usr/share/locale/XX/LC_MESSAGES alla (XX on keeletähis; nt et või de)

Hello.c

```
1     #include <libintl.h>
2     #include <locale.h>
3     #include <stdio.h>
4     #include <stdlib.h>
5     int main(void)
6     {
7       setlocale( LC_ALL, "" );
8       bindtextdomain( "hello", "/usr/share/locale" );
9       textdomain( "hello" );
10       printf( gettext( "Hello, world!\n" ) );
11       exit(0);
12     }
```

Allikas: http://oriya.sarovar.org/docs/gettext_single.html

Selgitusi

- setlocale() korjab keskkonnast kasutaja eelistused keelte ja kaasneva osas (kuupäevaformaadid, nädala algus, rahaühik jne)
- bindtextdomain() ütleb, et „hello“ nimelise keelekomplekti asukoha leiab /usr/share/locale alt (kuna vaikimisi, ei pea kirjutama)
- textdomain() ütleb, et keelekomplekti faili nimi on hello kõigis keeltes
- **gettext()** peaks ümbritsema kõiki stringe; alias _

Tööriistu

- gtranslator